



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/608,985	06/26/2003	Amitabh Srivastava	3382-64710	6401
26119	7590	08/16/2006	EXAMINER	
KLARQUIST SPARKMAN LLP 121 S.W. SALMON STREET SUITE 1600 PORTLAND, OR 97204			VU, TUAN A	
			ART UNIT	PAPER NUMBER
			2193	

DATE MAILED: 08/16/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/608,985	Applicant(s) SRIVASTAVA ET AL.	
	Examiner Tuan A. Vu	Art Unit 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 26 June 2003.
- 2a) ☐ This action is FINAL. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-36 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-36 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 26 June 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date: _____ |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date <u>5/24/2004</u> . | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is responsive to the application filed 6/26/2003.

Claims 1-36 have been submitted for examination.

Double Patenting

2. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

3. Claims 21, 25, 29 provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1, 14 of copending Application No.

11,330,053 (hereinafter ‘053), further in view of Srivastava et al., “Effectively Prioritizing Tests in Development Environment”, February 2002, MSR-TR-2002-15, Publisher: *Association for Computing Machinery, Inc* (hereinafter Srivastava). Although the conflicting claims are not identical, they are not patentably distinct from each other because of the following conflicts.

Art Unit: 2193

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

As per instant claims 21, 25, and 29, '053 claims 1, 14 also recite prioritizing test based on indicators indicating which portions are impacted by tests, portions including plurality of blocks wherein indication (i) indicates ones of the plurality of blocks are modified, hence this reads on marking changed logical abstractions; but '503 does not recite:

determining dependency information about binary files, propagating such information to determine subsystem and system dependency, marking changed and unchanged logical abstraction to prioritize tests.

Srivastava in a environment for obtaining list prioritized tests (see pg. 5-7) similar to '053 not only discloses using previous test results or data being marked to feed into new prioritized test, but also discloses binary files analysis to block system and subsystem, recording changes thereto in a propagation analysis hashing (L column, 4th para, pg. 3), and further obtaining dependency information (e.g. *along with ... coverage information* – L column 1st para, pg. 2; *changes ... propagated* – R column, 5th para, pg. 2; *list of modified blocks* – top para, R column. pg. 3; *marked as old block, impacted blocks, matching blocks* -- pg. 3, R column) so as to propagate it into graph elements in order to mark intra-procedural or interprocedural changes so to determine what are new block or unchanged blocks or potential impacted new block in view of a subsequent test coverage (see Fig. 1, Fig. 2, pg. 3-4); and producing prioritized lists (see top pg. 3).

It would have been obvious for one skill in the art to implement the prioritized tests by instant claims 21, 25, and 29 so that '053 steps of creating list of prioritized tests recording

Art Unit: 2193

modified or new blocks with respect to a given test version, which is analogous to Srivastava from above, be implemented to include using test results into a test optimization instance and effecting for which a dependency information determination and propagation via graph and hashing algorithm (see Srivastava) to enable marking of system and subsystem of abstractions in binary files as taught by Srivastava, because this enable appropriate marking as suggested in (i) leading to optimization on how to reduce paths in recurring tests for a given test version based on dependency information propagated in target binary files as Srivastava discloses (see Fig. 2)

Claim Objections

4. Claims 1-36 are objected to because of the following informalities:

The above claim recites 'software stored in memory comprising '*computer executable instructions for, determining ...; propagating ...; marking ...*'(line 3). The claim is reciting 'executable instructions' as for performing a list of one or more actions; and the use of *comma* after 'instructions for' is improper. Correcting this to be ':' is recommended as well as using one consistent type of punctuation mark (either a ',' or ';', not both) after each action thus listed.

Claims 1-36 exhibit language in terms like 'system' and 'subsystem' such as these are introduced without specifying a scope as to what *system* or *subsystem* these are all about. From the specifications, this can be system of abstractions as collected in the interface; or system of elements in test files. This indefiniteness in the language used is to be made more proper.

Appropriate correction is required.

Claim Rejections - 35 USC § 101

5. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. Claims 1-11, 6-31, are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The Federal Circuit has recently applied the practical application test in determining whether the claimed subject matter is statutory under 35 U.S.C. § 101. The practical application test requires that a “useful, concrete, and tangible result” be accomplished. An “abstract idea” when practically applied is eligible for a patent. As a consequence, an invention, which is eligible for patenting under 35 U.S.C. § 101, is in the “useful arts” when it is a machine, manufacture, process or composition of matter, which produces a concrete, tangible, and useful result. The test for practical application is thus to determine whether the claimed invention produces a “useful, concrete and tangible result”.

7. Specifically, claim 1 recites receiving a system definition and a request for dependency information, requesting via an application interface a dependency request comprising a target abstraction, and responsive thereto, receiving a dependency collection. As conveyed from the method steps above, the claim amounts to using a interface to create a request comprising some abstraction and obtaining in return a collection of dependency including more abstractions. There is no indication that the final collection (of logical abstraction) received is conveyed as being utilized in a real world useful application or transformed so as to create a tangible result via such application. That is, such collection appears to remain abstract and internal to what appears to an application interface process, internal for not being externalized in any form (e.g. as being displayed, as opposed to obtaining mere abstract data that remain internal to some process). The claim as a whole does not amount to an application that yields a real-world concrete, useful, and tangible result as required by the Practical Application Test. The claim is thus rejected for leading to a non-statutory subject matter.

Claims 2-11, 16-20 are also rejected for failing to remedy to the deficiency of the base claim.

Claim 27 recites the same steps of receiving, requesting and receiving steps as in claim 1; and 27-28 are also rejected for the same rationale as above.

8. Claims 21, 25 and 29 recite 'prioritizing tests based on either 'marked changes' or 'marked ... abstractions' (last line of claims). As a whole, the claims amount to a prioritizing action, the result of which is not conveyed as being exploited to yield a more tangible result, or explicitly expressed as some real-world result being perceived by the world making use of the method claimed or apparatus thereof. That is, the act of 'prioritizing' amounts to a mere abstract determination of order of some entities like test names, hence a concept not materialized in tangible result that would become useful when utilized via more actual action being performed.

Thus claims 21-24, 25-26, and 29-31 are rejected for leading to an Abstract idea, not fulfilling the Practical Application Test, and a non-statutory subject matter.

Claim Rejections - 35 USC § 112

9. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

10. Claims 1-36 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 1-36 exhibit the use of 'system' and 'subsystem' without specifying what system or subsystem this is all about. From the specifications, this can be system of abstractions as collected in the interface; or system of elements in test/binary files. This indefiniteness is to be corrected. Following are more representative improprieties.

Art Unit: 2193

Claims 1 and 27 recite the limitation "wherein the dependency collection comprises logical abstractions outside the logical abstraction's subsystem"(i.e. *logical abstraction's subsystem* - last line of respective claims). There is insufficient antecedent basis for this 'subsystem' imitation in the claim. Examiner will interpret this at best, as some 'logical abstraction' in a plurality thereof.

Claims 2-20, 28 are also rejected for not remedying to the above.

Claims 21 and 29 recite 'propagating dependency information to determine subsystem dependency information...' (line 2, line 4, respectively). There is insufficient antecedent as to how a *subsystem* is a *subsystem* when there is no prior mention of a definite system of a particular nature. Claims 21-24 and 29-36 are rejected for such lack of definiteness in elements claimed. This subsystem is treated as a subset of a larger test system or system of binaries or files.

Claim Rejections - 35 USC § 102

11. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

12. Claims 1-36 are rejected under 35 U.S.C. 102(a) as being anticipated by Srivastava et al., "Effectively Prioritizing Tests in Development Environment", February 2002, MSR-TR-2002-15, Publisher: *Association for Computing Machinery, Inc.*, pp. 1-10 (hereinafter Srivastava).

As per claim 1, Srivastava discloses method comprising:

receiving a system definition (e.g. Fig. 1; *Echelon, by the time ...all changes ... have been propagated* – 5th para, R col. pg. 2; *shifted addresses, different register allocation, small programs modifications* - 4th para, L column, pg. 3; *list of modified and new blocks* – R column, top para, pg. 3 – Note: Echelon infrastructure using Vulcan API as to receiving names of binary files and list of modification of blocks and *changes* to binary headers in terms of shifted addresses or small program changes reads on a application interface receiving system definition about binary files) and a request for dependency information (e.g. BMAT - 4th para, L column, pg. 3);

requesting via an application programming interface, a system dependency creation request comprising the received system definition, and a dependency request comprising a target logical abstraction (e.g. BMAT, API Vulcan, 4th para, L column, pg. 3 – Note: using API in Vulcan to effect a BMAT instance of files mapping result, using a set of binary files – via scanning from Fig. 1 – reads on request comprising a target logical abstraction – see *marked blocks... impacted blocks ...new blocks* – pg. 3, L column, top 2 para); and

receiving responsive to the application programming interface request, a dependency collection for the target logical abstraction comprising logical abstractions in one or more dependency chains with the target logical abstraction (e.g. Fig. 1, R column pg. 3 – Note: analysis of procedural graph to yield marking information depending on which to effect test coverage reads on abstractions having dependency chains),

wherein the dependency collection comprises logical abstractions outside the target logical abstraction's subsystem (e.g. *marked blocks... impacted blocks ...new blocks* – pg. 3, L column, top 2 para – Note: dependency of successor or predecessor in block analysis via

Art Unit: 2193

heuristic to prioritized list reads on abstraction within a subsystem and outside a subsystem – see Fig. 2, pg. 4, for iteration per sequence in and out current block).

As per claim 2, Srivastava discloses wherein the system definition is received as a file identification (e.g. *header files* - 5th para, R col. pg. 2; Fig. 1).

As per claim 3, Srivastava discloses wherein the system definition is received via a graphical user interface (e.g. *API ...for inspection*, 3rd para, L col. pg. 3).

As per claim 4, Srivastava discloses unchanged logical abstraction (e.g. *old blocks*, 2nd para, R col. pg. 3 – Note: old binary of Fig. 1 in matching reads on unmodified blocks or unchanged logical abstraction – see *logical level ... representation is symbolic*).

As per claims 5-6, Srivastava discloses wherein the dependency collection comprises logical abstractions dependent on the target logical abstraction; wherein the dependency collection comprises logical abstractions on which the target logical abstraction depends (*along with ... coverage information* – L column 1st para, pg. 2; *changes ... propagated* – R column, 5th para, pg. 2; *list of modified blocks* – top para, R column. pg. 3; *marked as old block, impacted blocks, matching blocks* -- pg. 3, R column).

As per claim 7, Srivastava discloses wherein the target logical abstraction is a changed logical abstraction (e.g. *new blocks*, 2nd para, R col. pg. 3 – Note: old binary of Fig. 1 in matching reads on modified blocks or changed logical abstraction).

As per claims 8-11, Srivastava discloses wherein the target logical abstraction comprises a basic block, a procedure, or a binary file; a basic block logical abstraction; one of a procedure logical abstraction or a binary file logical abstraction; a named object logical abstraction or a node logical abstraction (e.g. *procedures, blocks within each ... shifted addresses, different*

Art Unit: 2193

register allocation, small programs modifications - 4th para, L column, pg. 3; *list of modified and new blocks* – R column, top para, pg. 3; Table 1, pg. 4; *interprocedural graph* -R column, 4th para, pg. 3).

As per claim 12, Srivastava discloses displaying a representation of the collection comprising a number of affected (e.g. *impacted Blocks* Table 2-3, pg. 4, 6) logical abstractions.

As per claim 13-15, Srivastava discloses displaying a representation of the collection comprising a graphical presentation of a dependency chain; a representation of the collection comprising a graph of logical abstractions (e.g. *interprocedural graph* -R column, 4th para, pg. 3; Table 2-3, pg. 4, 6).

As per claim 14, Srivastava discloses displaying a representation of the collection comprising a list of logical abstractions (e.g. ch. 3.1, list of modified and new blocks – L col. top, pg. 3; Table 4, pg. 7).

As per claim 16, Srivastava discloses logical abstractions inside the logical abstraction's subsystem (e.g. set of sequences, CurrBlkSet – Fig. 2, pg. 4).

As per claims 17-18, Srivastava discloses means for displaying a proposed change risk (e.g. Fig. 4-5, pg. 5 – Note: percentage of impacted blocks over total sequences reads on display of proposed risk changes in view of proposed changes - *proposed* – top para, R column, pg. 2); i.e. percentage reads on displaying a change risk metric.

As per claim 19, Srivastava discloses wherein the relation is further adapted with a logarithmic function (see Fig. 5, pg. 5).

As per claim 20, Srivastava discloses computer-readable medium having executable instructions for performing the method of claim 1 (refer to claim 1).

Art Unit: 2193

As per claim 21, Srivastava discloses method comprising:

determining dependency information about binary files (e.g. *along with ... coverage information* – L column 1st para, pg. 2; *changes ... propagated* – R column, 5th para, pg. 2; *list of modified blocks* – top para, R column. pg. 3; *marked as old block, impacted blocks, matching blocks* -- pg. 3, R column - Note: using previous results or changes in files, and determining blocks for marking and marking insides binary files to find out about subsequent path on how the block test coverage is to be done reads on determining dependency information based on which to apply optimization algorithm upon runtime resources);

propagating dependency information to determine subsystem dependency information; propagating the subsystem dependency information to determine system dependency information (e.g. *changes ... propagated* – R column, 5th para, pg. 2 – Note: marking from using information changes to affected parts of a source binary as this is scanned under analysis reads on propagating this information determination or change information and use it for marking);

marking changed logical abstractions; marking unchanged logical abstractions dependent on marked changed logical abstractions in other subsystems (*marked as old block, impacted blocks, matching blocks* -- pg. 3 R column);

comparing test coverage to marked changed logical abstractions and to marked unchanged logical abstractions (e.g. *heuristic ...successor predecessor blocks* – pg. 3, R column ; *likely to be taken* – R column, bottom pg. 3; Fig. 2 – Note: based on marking information between old and new block, and applying heuristic thereto in order to predict how to skip path reads on comparing based on marked of changed and unchanged); and

prioritizing tests based on maximum test coverage of marked changed logical abstractions and marked unchanged logical abstractions (e.g. Fig. 2).

As per claims 22-23, Srivastava discloses coverage comprises tests for one subsystem (e.g. sequence Set – Fig 2 – Note: block set being tested based on weight of impacted blocks reads on subsystem of binary blocks), a subsystem among a plural subsystems.

As per claim 24, Srivastava discloses the test coverage comprises tests for plural subsystems and maximum test coverage is considered for marked changed logical abstractions and marked unchanged logical abstractions (pg. 4, L column; Fig. 2 – Note: impacted blocks based on marking of old and new blocks reads on test coverage for marked changed and unchanged – see R col., pg. 3) for said plural subsystems.

As per claim 25, Srivastava discloses a computer-based service comprising **means for** performing the same steps as recited in claim 21, namely:

determining binary dependencies for a defined system (e.g. *along with ... coverage information* – L column 1st para, pg. 2; *changes ... propagated* – R column, 5th para, pg. 2; *marked as old block, impacted blocks, matching blocks* -- pg. 3 R column);

propagating binary dependencies to identify binaries dependent on binaries in other subsystems and storing determined and propagated dependencies (*changes ... propagated* – R column, 5th para, pg. 2 - Note: marking from using information changes to affected parts of a source binary as this is scanned under analysis reads on propagating this information determination or change information and use it for marking);

marking changes and propagating marked changes using the determined and propagated dependencies; and prioritizing tests based on test coverage of marked changes and propagated marked changes;

all of which steps having been addressed in claim 21.

As per claim 26, Srivastava discloses marking proposed changes (e.g. *proposed* – top para, R column, pg. 2; *new version ... likely to be covered by a existing test* – 3rd para, R column, pg. 3 – Note: applying a test to a new code reads on proposed changes to an current test scenario).

As per claim 27, Srivastava discloses a computer-readable medium having executable instructions for performing a method comprising:

creating a system definition in response to receiving graphical user interface input (e.g. *Echelon, by the time ...all changes ... have been propagated* – 5th para, R col. pg. 2; *shifted addresses, different register allocation, small programs modifications* - 4th para, L column, pg. 3; *list of modified and new blocks* – R column, top para, pg. 3 – Note: Echelon infracstructure using Vulcan API as to receiving names of binary files – see Fig. 1 – and list of modification of blocks and *changes* to binary headers in terms of shifted addresses or small program changes reads on a application interface receiving system definition about binary files);

receiving a dependency information request via graphical user interface input (e.g. BMAT - 4th para, L column, pg. 3);

requesting via an application programming interface exposed by a dependency framework, a system dependency creation request comprising the system definition, and a target logical abstraction identifiable from the dependency information request (BMAT, API Vulcan,

Art Unit: 2193

4th para, L column, pg. 3 – Note: using API in Vulcan to effect a BMAT instance of files mapping result, using a set of binary files – via scanning from Fig. 1 – reads on request comprising a target logical abstraction – see *marked blocks... impacted blocks ...new blocks* – pg. 3, L column, top 2 para); and

receiving responsive to the application programming interface request, a dependency collection for the target logical abstraction comprising logical abstractions in one or more dependency chains with the target logical abstraction (e.g. Fig. 1, R column pg. 3 – Note: analysis of procedural graph to yield marking information depending on which to effect test coverage reads on abstractions having dependency chains);

wherein the dependency collection comprises logical abstractions outside the logical abstraction's subsystem (e.g. *marked blocks... impacted blocks ...new blocks* – pg. 3, L column, top 2 para – Note: dependency of successor or predecessor in block analysis via heuristic to prioritized list reads on abstraction within a subsystem and outside a subsystem – see Fig. 2, pg. 4, for iteration per sequence in and out current block; *interprocedural graph* - R column, 4th para, pg. 3).

As per claim 28, Srivastava discloses wherein the dependency collections further comprises logical abstractions inside (Note: dependency of successor or predecessor in block analysis via heuristic to prioritized list reads on abstraction within a subsystem and outside a subsystem – see Fig. 2, pg. 4, for iteration per sequence in and out current block; see *blocks within each procedures* – 4th para, L col., pg. 3) the target logical abstraction's subsystem.

As per claim 29, Srivastava discloses computer system comprising:

a processor coupled to volatile and nonvolatile memory; binary files stored in memory (see Binary - Fig. 1); software stored in memory comprising computer executable instructions for:

determining dependency information for binary files, propagating dependency information to determine subsystem dependency information, and propagating subsystem dependency information to determine system dependency information (refer to claim 21);

marking logical abstractions changed from a previous version (e.g. *Old Binary*, *New Binary* - Fig .1, R column pg. 3);

propagating marked changes according to the dependency information comprising marking unchanged logical abstractions dependent on marked changes in other subsystems (refer to claim 21);

comparing test coverage to marked changed logical abstractions and to marked unchanged logical abstractions; and prioritizing tests based on maximum test coverage of marked changed logical abstractions and marked unchanged logical abstractions (refer to corresponding rejection in claim 21).

As per claims 30-31, Srivastava discloses maximum test coverage is based on the total number of marked changed and marked unchanged logical abstractions touched (e.g. Fig. 2; L column, pg. 4) by a test system wide (Note: covering of subset or test sequence per iteration reads on system wide); wherein maximum test coverage is based on the sum of the total number of marked changed logical abstractions in a first subsystem touched by a test, and marked unchanged logical abstractions touched by the test in the first subsystem, wherein the marked unchanged logical abstractions depend on marked changed logical abstractions in other

Art Unit: 2193

subsystems (e.g. pg. 3, R column 2nd para, bottom para, pg. 3 to L column, pg. 4; ch. 4-5, pg. 4-5).

As per claim 32, Srivastava discloses user interface service (Echelon infrastructure using Vulcan API as to inspecting binary files – see Fig. 1- reads on application with GUI) comprising means for:

accepting a system definition comprising binary files (BMAT – pg. 3, R column, 4th para) in plural subsystems;

accepting an indication (ch. 3.1, pg. 3: ... *list of tests... of modified...blocks* – top R column, pg. 3; *hashing-based ... shifted addresses, different register allocation, program modifications* - pg. 3, L 4th para) of a target logical abstraction;

displaying dependency relationships between the target logical abstraction and a set of logical abstractions in binary files from two or more of the plural subsystems (e.g. Fig. 1; table 1-2, pg. 4; table 3-4, pg. 6-7).

As per claim 33-34, Srivastava discloses means for displaying a proposed change risk (e.g. Fig. 4-5, pg. 5 – Note: percentage of impacted blocks over total sequences reads on display of proposed risk changes in view of proposed changes - *proposed* – top para, R column, pg. 2); i.e. percentage reads on displaying a change risk metric.

As per claim 35, Srivastava discloses displaying a graph of relative risk for plural subsystems (Fig 3-5, pg. 5; Fig. 6-7, Table 3, pg. 6).

As per claim 36, Srivastava discloses means for displaying test coverage evaluation results (see Table 3-4, pg. 6-7).

Conclusion

Art Unit: 2193

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Tuan A Vu
Patent Examiner,
Art Unit 2193
August 14, 2006

Application/Control Number: 10/608,985

Page 18

Art Unit: 2193